automated database loader, and the administrator application. The Wind Tunnel Operations staff made a few feature requests, which were addressed and implemented within a week or two of receipt. During the LB435 test, values for 119 variables were collected at 19,684 points in time, resulting in approximately 2.2 million recorded values. The DARWIN database now holds close to 20 million values across 18 wind-tunnel tests.

Deployment of DARWIN version 2.5 at Ames allows wind-tunnel customers, including NASA and university researchers, the U.S. military, and commercial aircraft manufacturers, to access wind-tunnel test data from remote locations. Test data are available in near real time while tests are in progress and can be compared against archival tests already in the database. The DARWIN system lowers the cost of wind-tunnel tests by reducing travel requirements and providing rapid data analysis capabilities. Advances made in the version 2.5 release facilitate DARWIN support of the Unitary Plan Wind Tunnels and the National Full-Scale Aerodynamics Complex, and also pave the way for adoption of the DARWIN system at NASA Langley Research Center.

Point of Contact: David Korsmeyer
(650) 604-3144
dkorsmeyer@mail.arc.nasa.gov

Remote Large Data Set Visualization

David Ellsworth

Simulations run on large parallel systems produce data sets that contain hundreds of megabytes to terabytes of data. The researchers producing these data sets prefer to visualize them using their personal workstations. High-end PC workstations currently have the computation and graphics power to perform these visualizations. However, these workstations do not have sufficient memory to completely load large data sets. This research, which enables the use of personal workstations for visualizing data sets that are too large to be stored on personal workstations, will increase the productivity of researchers who are pushing the limits of simulations by producing very large data sets.

Because personal workstations have limited memory, out-of-core visualization techniques must be used. These techniques calculate the visualization with only a fraction of the data set resident in memory. In addition, many data sets are so large that they can only fit on central file servers. Since most file servers do not have significant extra central processing unit (CPU) and memory capacity, remote out-of-core visualization is required.

Our earlier research developed an out-of-core visualization technique called application-controlled demand paging. This technique loads the data required by the visualization algorithm from disk into main memory as necessary. The technique works well because most visualization algorithms only use a small fraction of the entire data set. The overall speed typically increases as the user interacts with the data set, since previously loaded data are retained if possible. This means that the overall speed will soon approach the speed that would be seen if the entire data set were loaded into memory.

However, the original implementation of out-of-core visualization using demand paging did not try to perform computation, disk access, or network access at the same time. When the implementation detected that data must be read, it stopped the calculation and waited for the data to be read from disk. If the data was read from disk on a remote server, the network transfer time increased the delay.

The new algorithm overlaps the computation, disk access, and network transfer by dividing the visualization into a number of tasks. When one task must wait for data to be retrieved from disk, a scheduling algorithm runs another task to keep the processor busy. The data retrieval from local or remote disk proceeds independently while the requesting task waits.

The new algorithm is general enough to support a variety of visualization techniques. The visualization techniques do have to be modified so that the work can be divided into a number of smaller tasks. However, this is the same as modifying the visualization so it can be run in parallel, which is useful in itself and is likely to have already been performed.

Figure 1 shows one data set that was visualized using the new algorithm. This figure shows the Harrier jet flying slowly 30 feet off the ground with spherical particles showing the path of the jet exhaust. This large (107 gigabyte, 1,600 time-step) data set was produced by Ames researchers during their investigation of the cause of oscillations seen during landing.

Figure 2 has timings that show the improved performance of the new algorithm. It shows the time required to compute the entire 1,600-frame animation when data were read from a remote system over a 800-Mbit/second HIPPI network. Both the local and remote systems were SGI Onyx systems. The chart compares the performance seen using the standard

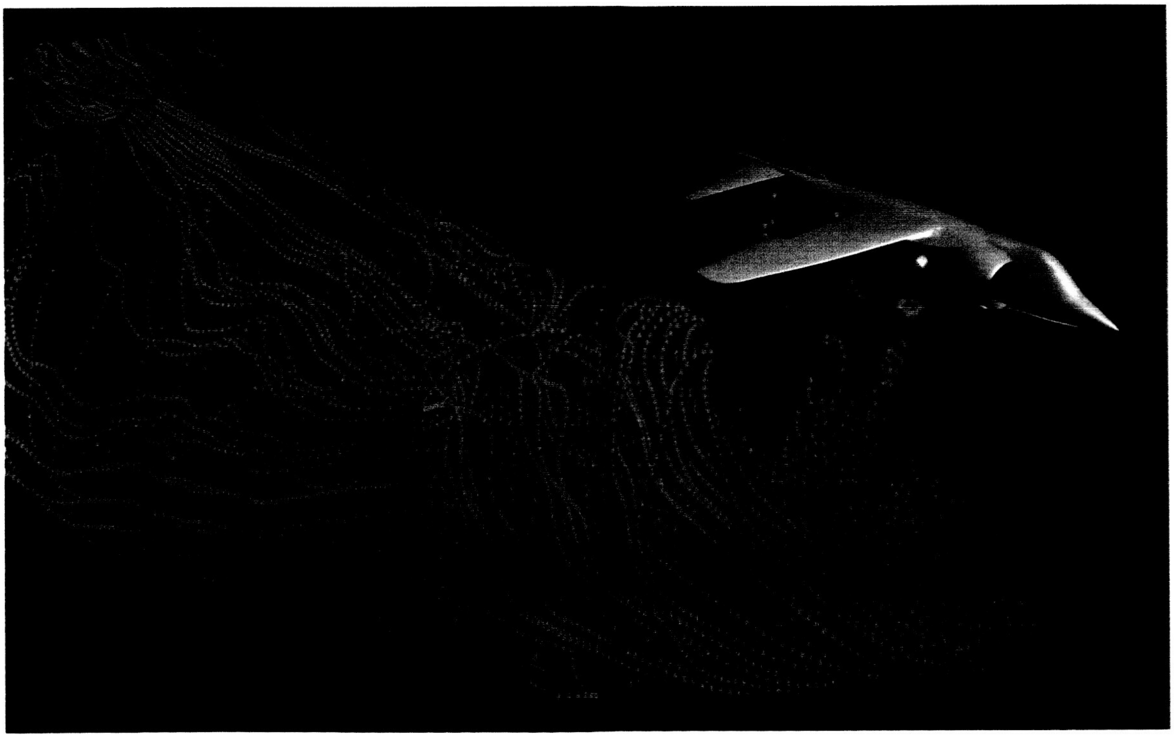

Fig 1. Simulation of the Harrier flying 30 feet above the ground.

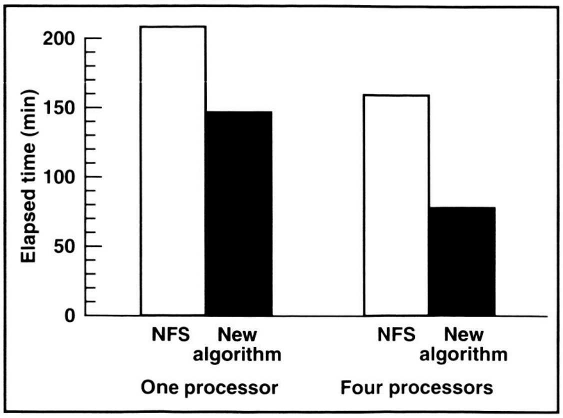

Fig 2. Improved performance of the new algorithm compared to using NFS, for one and four processors.

Network File System protocol to retrieve the data against the time using the new algorithm. When using one processor, the time decreased from 207 minutes to 146 minutes, a 30% decrease. When four local processors were used, the time decreased by more than half, from 159 to 77 minutes.

Point of Contact: David Ellsworth
(650) 604-0721
ellsworth@nas.nasa.gov

Computational Simulation of High-Lift Wind-Tunnel Test

Stuart E. Rogers

The modeling requirements for validating viscous computational fluid dynamic simulations of a high-lift trapezoidal wing were investigated. This wing has a full-span slat and a full-span flap, and has been tested extensively in the Ames 12-Foot Pressure Wind Tunnel. Because of the size of the wing, there are significant facility effects in the data from the 12-foot wind tunnel. The objective of the work was to quantify the effect of wind-tunnel walls, and to develop a computational model that would effectively simulate this effect.

Two different computational models of the test facility—differing in fidelity—were developed and tested. The first of these is a complete viscous simulation of all surfaces within the 12-foot wind-tunnel test section. This computational model more than doubles the cost of computing the flow over the trapezoidal wing, as compared to simulating free-air conditions. The second model of the wind-tunnel test section was a highly simplified straight duct, whose wall was treated as an inviscid surface. The surfaces used in this second model are shown in figure 1. Simulations with the second model required the same computational time and memory as a free-air simulation.

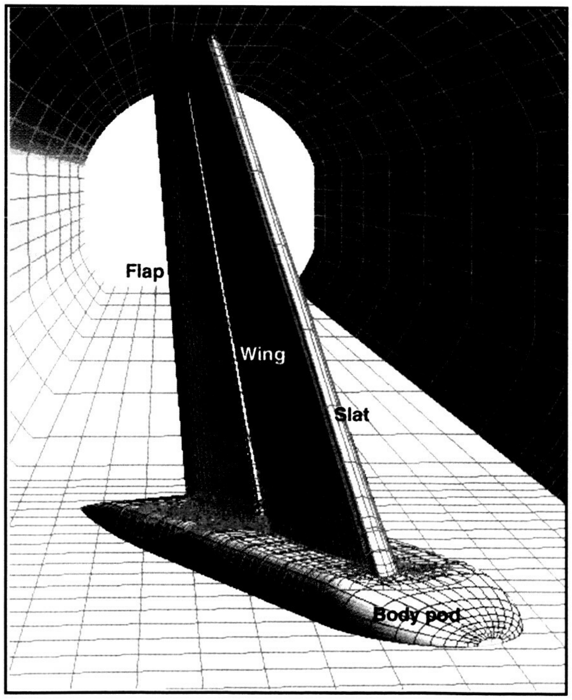

Fig. 1. Surfaces of the trapezoidal wing and simplified wind-tunnel test section.